NISTIR 5631

An Analysis of ANSI ASC X12 and UN/EDIFACT Electronic Data Interchange (EDI) Standards

Robert Aronoff
Karen Hsing

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Gaithersburg, MD 20899

NIST

NISTIR 5631

An Analysis of ANSI ASC X12 and UN/EDIFACT Electronic Data Interchange (EDI) Standards

Robert Aronoff
Karen Hsing

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899

April 1995

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Arati Prabhakar, Director

Table of Contents

List of Figures

List of Tables

AN ANALYSIS OF ANSI ASC X12 AND UN/EDIFACT[1] ELECTRONIC DATA INTERCHANGE (EDI) STANDARDS

1. Background

How can we increase our efficiency, reduce expenses, do the job better, and offer greater value to our customer? These are the perennial questions that success-oriented enterprises ask of themselves. Commercial and non-commercial ventures, alike, are always looking for ways to increase productivity, reduce costs, provide increased added-value, and become more competitive and efficient.

In today's world, application of new technology is an approach frequently taken when trying to increase an organization's productivity and capability, and reduce costs. Perhaps, the appeal of new technology derives from, among other things, the fact that it offers new and additional capabilities, avoids the problem of having to refine current methodology in which some may have a vested psychological interest, is highly visible, and provides the inherent appeal of novelty and a change from the current approach. Among the new tools made available in recent years and most often employed, are those deriving from the explosive growth of computing and networking technologies.

Many business related activities have been, and continue to become, computerized. However, this conversion to electronic processing is not without its difficulties, and can be a two-edged sword. While offering the potential to improve quality and efficiency, computerization can sometimes produce the opposite effect if processes and information cannot be well integrated. Computerized applications can become isolated islands of automation, thus causing coordination of operations to actually become more difficult and less convenient rather than easier and more efficient. If multiple computerized applications are intended to interoperate, these applications must have the ability to share information among themselves, have a common understanding of that shared information, provide a common format for the information, communicate with one another, and perform related/coordinated activities while remaining fully within the computerized, electronic mode.

---

[1]ANSI ASC X12 (American National Standards Institute, Accredited Standards Committee X12)

UN/EDIFACT (United Nations Electronic Data Interchange For Administration, Commerce and Transport)

2. Introduction To EDI Concepts

The question arises for the forward thinking, success-oriented enterprise: how can technology be applied to improve current business practices? It turns out that many sectors of the economy, (including, for example, the health care industry), in addressing this question, have arrived at very similar conclusions. Most enterprises deal with some form of information which is commonly understood and interpreted by business partners and which must be processed by, and communicated between, those participating partners. In particular, there seems to be a consensus that considerable improvements can be realized (including, for example, cost savings, productivity increases, and the enabling of increased capabilities and functionality) by computerizing, machine processing, electronically transmitting business communications and transactions, and, in general, automating the business procedures. The application of this technology can lead to increased speed in processing, reduction of redundant operations and information, greater control over processes, improved audit control, and additional functionality. The name given to this general approach to these types of business activities is Electronic Commerce (EC). That part of EC that concerns the actual interchange of business transactions is called Electronic Data Interchange (EDI).

In the traditional business setting, many aspects of the way data is represented, stored, and interchanged, are taken for granted. For example, the local language (e.g., English/American) often provides the syntax and semantics for the data, while paper, file drawers, and the postal service provide the means for storage and transfer of the information. The commonality of these underlying representation and communication tools makes the traditional approach highly flexible, upgradable, often transparent, and, therefore, of little apparent concern.

EDI is intended to provide an alternative approach to the traditional business model of information interchange and to, in fact, improve upon it. The "paperless office" is the logical extension of this approach. As the name EDI implies, the data is to be stored, and the interchange is to be accomplished using electronic media. In order to accomplish this end, the following things must be achieved: the computerization of information, electronic storage of data, automatic parsing and interpretation of data, electronic and automatic processing of data, and the transfer of data electronically.

If we look in detail at the requirements, we find that two general categories are sufficient to characterize most of what must be addressed in order to realize interoperable EDI. First and most fundamentally, there is a need for a common information model and definition of the information involved in interchanges so that both/all parties involved have a uniform view, and understanding, of the information being exchanged. What is really needed is the ability to represent the information to be interchanged in a language which is mutually agreed upon and understood by all participants who are party to the interchange. In natural language, this is achieved by a common syntax and semantics for the information being exchanged. In the computerized world of EDI, this is achieved in a similar fashion.

As an example, an English speaking cardiologist is able to collaborate with, and be well-understood by, another English speaking cardiologist because they communicate with a common language, both syntactically and semantically, and they reference a common body of knowledge. In fact, this communication is done rather effortlessly and generally transparently, with little attention to the communication process itself. However, as more elements of dissimilarity get introduced into the communication activity, the process itself becomes increasingly noticeable and difficult and requires greater attention. That is to say, when the English speaking cardiologist must communicate these same ideas to a French speaking art teacher, the exchange of information takes on a much different tone, requires a lot more effort, is generally not done transparently, and may, in fact, not even be fully successful if enough dissimilarity exists in the language and knowledge bases of the participants.

Such a scenario points out the need to establish both a common syntax and semantics to be selected and used by communicating partners in EDI transactions, as well as a common knowledge base, if such a system is to enable true interoperability. Nevertheless, while in the ideal situation it may be desirable for both/all parties involved in the interchange to directly use the same syntax and semantics, variations from these ideal conditions can still permit interoperable communication if mechanisms are available (e.g., a translator) to provide for reliable and faithful conversion from one party's view to that of the other party.

Once the ability to represent and understand the interchange information has been provided, a second category of concern then reaches prominence. That is, once agreement is reached concerning what is to be communicated and how it is to be represented, the need arises for a common method of transfer of this information which has been represented in this common format. It is the combination of the ability to transfer this information electronically with the ability to understand and process this information in an automated fashion which constitutes the full extent of the advantage of this technology.

In order to attempt to adequately present the issues, this paper limits its focus to a discussion of how to provide a common view of the data involved in the interchanges. In particular, we will study the elements involved in providing a common syntax and semantics for the representation of communications between parties. The process of the actual transfer of this information, once appropriately represented, will not be discussed in this paper.

At the present time and in the foreseeable future, two standardized approaches have come to dominate the information representation aspect of EDI technology (i.e., the ANSI ASC X12 standards and the UN/EDIFACT standards). In this paper we will look at the general approach common to both these standards in providing syntactical and semantic aspects of EDI, and we will provide an analysis of those differences which occur between these standards regarding the ways in which these two standards approach their solutions to providing a common information model.

3. Overview of EDI Standards

3.1 Organization of EDI Standards

The EDI standards, ANSI ASC X12 [1,2] and UN/EDIFACT [3,4,5], are quite similarly organized.

1) First, these standards specify a way to define and structure information. In particular, they specify a standard format/syntax for structuring business information for electronic interchange. This includes a set of design rules which govern how to specify each of the components in the standard format and how these components are organized and inter-related.

2) Secondly, these standards provide directories which maintain ordered collections of the message components once they have been defined in accordance with these rules. At present, transaction sets (also called messages), numbering in the hundreds, have been specified and assigned to these directories.

Each of these standards requires only a small number of basic building blocks to enable these interchanges. ANSI ASC X12, for example, offers the following six basic components: 1) interchange envelope, 2) functional group, 3) transaction set, 4) data segment, 5) data element and 6) loop construct that allows repetition and nesting of segments. The most encompassing of these building blocks supports the concept of the interchange itself. The interchange comprises the entire package which is exchanged between trading partners. Within the interchange, then, are contained the actual transaction set(s) which organize the information and which, in turn, contain the collection of individual and composite pieces of information in the form of data segments and their constituent data elements.

Control structures are used to compose, organize, and delimit these basic building block structures. In order to be able to parse the information exchanged between business partners, there is a need to identify the beginnings and endings of the message and its parts, and also to carry necessary control information for addressing, processing, and checking completeness of the enclosed information. The primary control structures available to provide these functions include header and trailer segments for building blocks such as interchanges, transaction sets, functional groups, and security information. In a like manner, loop control header and trailer segments can be used to enclose a body of information where explicit demarcation is needed to disambiguate between multiple occurrences of repeated information within loops.

In contrast to these larger data units just described, the smaller elemental data units comprising the messages are delimited by the use of separation and termination characters which are either known a priori to the trading partners or are identified among the partners by specifying them within the appropriate control structure. In these cases, all that is necessary is to provide well-defined characters to be recognized during the parsing operation to identify divisions between

information elements. The control information is already provided for in the control structures of the enclosing data structures.

3.2 Scope of Applicability of EDI Standards

The scope of applicability of these standards is rather extensive. The design rules and standard format specified in the X12 and UN/EDIFACT standards are not limited to one sector of the economy, but, rather, are applicable across industry types and are useful for firms of different sizes and for a wide variety of purposes. To date, messages or transaction sets, based on these syntax rules, have been defined for a broad range of applications, including: health care, banking, customs declaration, order processing and purchasing, transportation, and education.

As the rest of this paper will suggest, the two sets of design rules and formats specified by these two standards (X12 and UN/EDIFACT) are quite similar. Where they do differ, those differences can generally be attributed to variations in business viewpoints of the standards developing committees and/or of the intended user communities. The ANSI ASC X12 standards are American national standards which are widely deployed in North America. The UN/EDIFACT standards, on the other hand, are international standards that are used worldwide.

3.3 An Example Scenario: How EDI Can Be Used To Implement A Traditional Business Process

This section describes an example business procedure as a way of introducing the appropriate EDI terminology and concepts needed to convert such a process from the traditional paper flow to the target EDI environment.

Let's look at what might occur at a physician's office after services have been rendered to patients and when, at some periodic interval, the paperwork has to be submitted to the appropriate payment organization for reimbursement. In a traditional paper-oriented environment, the physician's office often sends an envelope containing a collection of relevant forms (documents), such as health care claims and claim status inquiries, to an insurance company (a trading partner in EDI terminology). Generally, some prearranged ground rules have been set, either explicitly or implicitly, between the trading partners (in this case, the physician's office and the insurance company), in order to assure that the appropriate information is exchanged and understood by both partners. In EDI terminology, these explicit agreements are called interchange, or trading partner, agreements and they set the context for the interchanges. These agreements may be generic ones, to which any trading participant agrees to adhere, or they may be individual bi-lateral or multi-lateral agreements, which are more highly customized and individualized.

For efficiency and ease of processing, the claim forms and the inquiry forms will often be separately grouped. Other forms may be grouped together because they contain related information. Each form usually has a form name and/or a form number which identifies the form type and provides a context for the information contained therein. Each type of form is organized

in a particular manner with various specific fields capable of holding information whose values are interpreted within the context of the particular field in which the value is placed. In the claim form, for example, there might be sections such as: patient identification and demographics, diagnosis, and treatments and charges. Furthermore, each section might contain one or, possibly, more items. For example, the patient demographic section could contain the patient's name, address and phone number. A single line might be provided for each of the name and phone number information entries, while several lines would usually be provided for the address, so as to designate a line each for street address, city, and state information. Additionally, this information may be repeated many times to present claim information for several patients of a given doctor. If the office activities of several doctors are all serviced by the same administrative provider, the administrator may collect claims from each of these doctors and their patients, and organize this submission envelope into groups of claim forms and status inquiries for each of the doctors.

What results, then, is a submission package with the following structure and contents. Viewing this package from the outside and gradually peeling away the layers to reveal the nesting structure of the information, we first see the envelope, itself. In addition to containing all the information to be exchanged, the envelope identifies who is sending and who is to receive this information. It also contains any necessary information to identify the manner in which the envelope is to be delivered (e.g., the postage type and amount could indicate postal delivery of a particular class, perhaps requiring confirmation upon receipt, or expedited delivery guaranteed by a certain time).

Assuming that we have an efficiently organized sender, the contents of the envelope will be organized to group like transactions. For example, if there are multiple departments within the destination organization to which different transactions are to be sent, then subordinate groupings of transactions may be made based on the destination department. Within each of these groupings, there can be further groupings by doctor, with each patient entry being associated with the appropriate doctor. The individual patient claim forms, then, constitute the next level of nesting. And within each of these claim forms there are sequences of information fields which, in turn, consist of simple or complex data fields filled in with, hopefully, appropriate values.

To illustrate the point that the successive nesting of this information provides meaningful groupings of information and facilitates the transfer and understandability of this information by the end systems, let us reverse direction and look at the submission package from the inside looking outward. What we see is consecutive aggregations and groupings of information elements from elemental data items, through the composite data items, to the functional grouping of information, to the ultimate submission package or interchange. Each level of aggregation serves to group related information together and provide semantic context for the information as well as to enable more efficient distribution and processing of the information.

3.4 EDI Terminology

In EDI terminology, each line of information, such as the name or the phone number of a patient, comprises a data element. As the name suggests, the data element is the most basic, or elemental, unit of information upon which these information exchanges are based. Larger units of commonly related information can also be defined. There are generally two ways to accomplish this aggregation of elemental information (the composite data element, and the data segment), both of which are comprised of multiple data elements. Composite data elements are intermediate units contained in data segments, whereas data segments are intermediate units contained in even larger units, transaction sets. If an address is defined as a data segment, each individual piece of information, such as "city", is called a data element. If, on the other hand, an address is defined as a composite data element, then "city" becomes a component data element. While having considerable similarities, these aggregated structures tend to be used in different situations depending on the degree of close coupling of the information represented, the degree of flexibility needed in repetition of the information, and the need to dynamically represent new data types without statically producing a new definition.

For example, an address represents a higher order concept consisting of multiple parts and can be represented by an address segment or by a composite data element. The choice of which one to use has more to do with the specific syntax rules of the relevant standard and how the information is intended to be used than it does with the conceptual nature of the aggregated information. While some have questioned the need to have both of these aggregating mechanisms and prefer to only use the data segment mechanism, there are some differences which may be useful from time to time. The composite data element must explicitly enumerate all possible repetitions of its components and, therefore, would tend not to be used when the same component is repeated many times. The data segment, on the other hand, which is included in transaction sets, can be contained in a loop mechanism which permits increased repetition without explicit enumeration of each instance. (See Appendix B of [6] for a detailed comparison of data element and composite data element usage in both standards.)

The data segment structure consists of logically related data elements in a defined sequence. In this example, the patient identification and demographics section may correspond to an EDI patient identification and demographics segment where the patient's name and phone number are data elements within it and the address segment is a nested segment within the patient demographics segment. The collection of related data segments corresponding to the three sections in the example health care claim form, together, are referred to as a "transaction set" in X12 (or a "message" in the UN/EDIFACT standard). Within each transaction set specification is a table which indicates which segments must be used (i.e., are mandatory), which segments may be used (i.e., optional), and the order and permissible number of repetitions of the segments. These transaction set specifications can be found in the transaction set directory of the appropriate standard.

Another grouping construct is also provided to enable aggregation of related elements in EDI.

This construct, the functional group, which will be explained in more detail later in the paper, is somewhat differently applied in X12 from the way it is used in UN/EDIFACT. The "functional group" can be viewed as analogous to a paper-clipped pile of forms in the paper world. It enables the grouping of similar transaction sets/messages. An EDI interchange, therefore, can contain one or more transaction sets/messages or one or more functional groups and the interchange, itself, represents the entire package or envelope being exchanged, and, therefore, is analogous to a postal service envelope.

Figure 1 attempts to graphically represent the nature of the grouping capabilities and intent of the EDI construct.

3.5 ANSI ASC X12 Data Structures

Working inward from the outermost data structures involved in organizing a message interchange, the first three EDI data structures that we look at in more detail are organized in a similar fashion. They each include beginning and ending segments which clearly delimit the data structure. The interchange envelope, functional group, and transaction set EDI data structures are each formed by enclosing their contents between a pair of control segments called the header and the trailer. These control segments are comprised as follows:

- The start of the interchange envelope is designated by the interchange header (the ISA[2] segment), and is terminated by an interchange control trailer, (the IEA segment). The ISA segment contains data elements that specify how many transaction sets are in the interchange envelope, who the sender is, and the destination of the interchange. The IEA segment contains data that helps the receiver determine if the transmission is complete and if all the data in the interchange envelope has been received.

- The functional group begins with a header (the GS segment), and ends with a trailer (the GE segment). The GS segment contains data that identifies the type of transactions contained in the group, the sender and the receiver's application codes, the transmission date and other information such as the version of the standard that is being used. The GE segment specifies the number of transaction sets contained in the group, and the control number assigned to the group by the sender. The control number in the GE segment can be used by the receiver to check for a match with the control number in the GS segment to ensure that the information in the group has been completely received.

- The transaction set begins with a header (the ST segment), and ends with a trailer

[2] Both ANSI ASC X12 and UN/EDIFACT use 2 or 3 letter identifiers to identify their data segments. These are identifiers, not acronyms. Therefore, these identifiers do not have any intrinsic meaning. Their meaning comes from their association with the data segments they are intended to identify.

(the SE segment). The ST segment contains the number of segments included in the transaction set and the control information assigned by the sender. The SE segment contains control information used for verifying that all the data in the transaction set has been received.

Contained within the above described structures are progressively more refined data structures.

The following structures differ from the three described above in that they do not have separate control data segments to signal their beginnings and endings or to carry along additional control information. Rather, these structures (the data segment, data element, and one form of loop) use a simpler mechanism for signalling their starts and endings.

- Each segment has a unique identifier (the segment tag), that comprises the first three bytes of a segment. In a similar fashion, the end of each segment is signalled by a single byte segment terminator. The particular character used for segment termination is specified by the interchange sender in the interchange header. Except in certain well-specified situations, most individual segments may be repeated, and groups of segments may be repeated in loops.

- Data elements are the basic information unit. In the X12 standard, for instance, eight types of data elements are defined. These include: numeric, decimal number, identifier, string, date, time, binary, and fixed-length string. To identify data elements in the data stream, each data element (simple, or composite), in a data segment is preceded by a one byte separator. If the data element is a composite one, each component data element within it is additionally preceded by a one byte subelement separator. The characters used for the data element separator and the subelement separator are specified by the interchange sender in the interchange header.

  Unlike identification of the more encompassing data structures such as the transaction set and the data segment, data element identifiers are not included in the transmission of data elements within either composite data elements or data segments. Rather, data elements are recognized by their specifically assigned positions within a data segment. Optional elements not sent are indicated by including the separator associated with that element. The use of this position dependent mechanism to eliminate the need to send data element identifiers, conserves transmission bandwidth and enables proper message parsing while still supporting optionality of data elements.

- Loops are available in two flavors, bounded and unbounded. A bounded loop unambiguously delimits the loop by beginning the section to be repeated with a loop header (the LS segment), and ending that section with a loop trailer (the LE segment). Unbounded loops, in contrast, do not contain explicit control header

and trailer segments. Rather, there are specific rules which are designed to help avoid ambiguous situations (e.g., the first segment of an implicit loop must appear once and only once in an occurrence and shall not appear elsewhere in the loop). Either form of loop may be nested within another loop.

The following figure, Figure 1, depicts the hierarchical structure among X12 EDI components, just described.

```
ISA     Interchange Control Header
        GS      Functional Group Header
                ST      Transaction Set Header
                                .
                        data segment
                                .
                                .
                        LS      Loop Header
                                        .
                                        .
                                data segment
                                        .
                                        .
                                LS      (nested loop) Loop Header
                                                .
                                        data segment
                                                .
                                                .
                                LE      (nested loop) Loop Trailer
                        LE      Loop Trailer
                                        .
                                        .
                                data segment
                                        .
                                        .
                SE      Transaction Set Trailer
        GE      Functional Group Trailer
IEA     Interchange Control Trailer
```

Figure 1. X12 EDI Structure

3.5.1 ANSI ASC X12 Security

X12.58, the security structures standard, defines data formats to support authentication and encryption in order to provide data integrity, confidentiality, and user authentication. In a like manner, the appropriate security trailer segment (either S1E for the functional group, or S2E for

the transaction set), is used immediately preceding the segment terminating that given level (i.e., the GE segment for the functional group or the SE segment for the transaction set). To accommodate different user needs, these security features are offered at two levels, i.e., both at the functional group and at the transaction set level.

At each of these levels, security features (i.e., authentication and encryption) are optional and independent of security at any other level. If security is desired, the security header segment, (i.e., S1S for functional group level and S2S for transaction set level), is used immediately following the segment initiating the beginning of the appropriate level (i.e., the GS segment for the functional group, or the ST segment for the transaction set) Likewise, the security trailer segment, (S1E for the functional group and S2E for the transaction set), immediately precedes the segment terminating the level (GE or SE, respectively). Sometimes, to provide a greater level of security or to address particular business requirements where processing of transaction sets requires greater, or different, security policy than the processing of information in the functional group, the security features of both the functional group and the transaction set can be used. If both levels are desired, the sequence of segments would be as follows:

```
ISA
        GS
        S1S     Security header level 1
                ST
                S2S     Security header level 2
                        (The transaction set segments)
                S2E     Security trailer level 2
                SE
                        (Other transaction sets using the same or a non-secured format)
        S1E     Security trailer level 1
        GE
                (Other functional group using the same or a non-secured format)
IEA
```

Figure 2. X12 EDI Structure with Security Functions

In X.58, the security header and trailer for both levels are defined with identical formats. In the header, first, the security originator name and security recipient name need to be filled and then security type needs to be set to indicate the combination of authentication or encryption to be used and the algorithm to be employed. If authentication is being used, the authentication key name and authentication service code must be inserted. If authentication is not to be used, these optional fields are set to empty and the trailer segment is set to blank. If encryption is to be used,

the encryption key name and encryption service code must be inserted. If encryption is not to be used, all data elements and their preceding data element separators are omitted.

3.5.2 ANSI ASC X12 Acknowledgment

If the preparer of the interchange desires an acknowledgment to be returned from the recipient, then he/she can indicate that request for acknowledgment in the interchange header. Likewise, if no acknowledgment is desired, then the sender so indicates and neither the recipient nor any intermediate network service provider is to return an acknowledgment. If an acknowledgment is requested, then the recipient must return an acknowledgment. If, for some reason, the original interchange transmission could not be delivered, an interchange acknowledgment may be returned to the original sender by a network service provider indicating that delivery was not successful.

The interchange acknowledgment segment, TA1, is used to acknowledge one interchange header and trailer envelope. The TA1 segment reports the success of processing a received interchange envelope or the non-delivery by a network provider. In an interchange envelope, a TA1 or multiple TA1s are placed after the interchange header and before the first functional group or before the interchange trailer if there are no functional groups. More than one interchange acknowledgment may be placed after the interchange header, provided the sender and receiver ID values are appropriate for the proper delivery of the acknowledgment. The interchange control number in the TA1 segment must be the same as that in the ISA segment for which the acknowledgment was prepared. The control number serves as the link between the interchange envelope and the acknowledgment of that interchange envelope.

The interchange acknowledgment, however, does not report any syntactic analysis status of the functional groups contained in the interchange envelope. To report this status, the functional acknowledgement transaction set, 997, is used. Only a single 997 response is allowed per functional group. It reports on the syntactic integrity of the entire received group of transaction sets. In the 997 transaction set, acknowledgment information containing error codes, if applicable, is provided for each of the segments and data elements contained in a transaction set. The acknowledgement information for each transaction set within a functional group appears in the same order as the original transaction sets appeared in the functional group that was received and is now being acknowledged.

3.5.3 EDI As Applied To A Particular Application

The health care claim, in general, serves as an invoice and notice of services performed. The function of the health care claim is to provide to the payer all information necessary to determine the amount to be paid to the provider for the health care services rendered, and to track encounter and other medical information related to the patient and the provider.

The ANSI X12 health care claim transaction set 837 is a variable-length record designed to allow submission of health care claim billing and/or medical encounter information, from providers of

health care services to payers. Information contained in this transaction may be integrated electronically into the payer's processing system, thus allowing an automated exchange and processing between the provider's system and the payer's system. Information contained in this transaction may be exchanged among primary, secondary and/or tertiary payers, if coordination of benefits is required.[1]

Physicians, hospitals, pharmacies, dentists, and other medical facilities or suppliers, are among the list of possible providers of health care products or services submitting the 837 transaction set. A subscriber is the person in whose name the insurance is purchased. In the case of individual coverage, the subscriber and the patient would be the same. However, in the case of family coverage, for example, the patient could differ from the subscriber (e.g., the subscriber might be the parent, while the patient might be the child). The payer is an organization that pays claims or administers the insurance product or benefit or both. For example, a payer may be an insurance company, health maintenance organization (HMO), preferred provider organization (PPO), government agency (e.g., HCFA), an entity such as a third party administrator or third party organization that may be contracted by one of those groups.

The common way of defining a transaction set definition is to specify the definition in the form of tables, which generally serve three different purposes: i.e., heading, detail, and summary. The heading, detail and summary areas of a transaction set are usually referred to as tables 1, 2 and 3, respectively, and are used for logical grouping of the transaction set elements. Both the detail and summary areas are optional. In the X12 standard, the 837 transaction set specification contains two tables: a heading table and a detailed table.

In the 837 transaction set, the heading table contains a single data segment, the submitter's information segment, which contains information about the submitter who prepared the specific claim (e.g., the submitter's ID, address, contact phone number and FAX number). The detail table contains claim information organized in repeated and nested segments. Figure 3 depicts, in a relatively self-explanatory way, a limited expansion of the loop repetitions and nesting of the overall structure of this detail table. The information contained in the claim segment includes such data as: orthodontic information, tooth summary, disability information, ambulance, chiropractic, therapy, medical equipment and medical procedure information. The service line segment, which provides one part of the claim information, contains information regarding drugs, physicians (e.g., the attending, the operating, and the ordering physician), and accommodation information. In the insurance segment, constituting the other part of the claim information, multiple primary and secondary insurance information can be specified. For fully detailed definitions of these various segments and data elements, see the X12 standard.

- Provider
  - subscriber
    - patient
      - claim
        - service line(s)
        - insurance
      - claim
        - service line(s)
        - insurance
    - patient
      - claim
        - service line(s)
        - insurance
  - subscriber
    - patient
      - claim
      - claim
- Provider
  - subscriber
    - patient
      - claim
        - service line(s)
        - insurance

Figure 3. Structure of Table 2 from Transaction Set 837

3.6 UN/EDIFACT Interchange Format and Control Structures

A UN/EDIFACT interchange has the same basic structure as does an X12 interchange. The syntax rules for UN/EDIFACT are similar in many ways to the syntax rules for X12. As shown in Figure 4, the header and trailer segments are used to designate the interchange envelope, the functional group, and the message. As in X12, a UN/EDIFACT message is made up of segments which are themselves made up of data elements.

In UN/EDIFACT, the term "message" is used rather than "transaction set" as in X12. Another difference in nomenclature is that the control segments sometimes are called the service segments in EDIFACT. There is, in UN/EDIFACT, one additional control/service segment not found in X12, the Service String Advise (UNA) segment. This optional UNA segment, if used, is the first

segment in an interchange and is used to specify delimiters and describe the character set that is being used in the interchange. There is also an optional service segment in UN/EDIFACT called the UNS segment, which can be used to divide a message into the header, detail, and summary sections.

```
UNA  Service String Advise
     UNB  Interchange Header
          UNG  Functional Group Header
               UNH  Message Header
                    .
                    .
                    User Data Segments
                    .
                    .
               UNT  Message Trailer
          UNE  Functional Group Trailer
     UNZ  Interchange Trailer
```

Figure 4. EDIFACT Interchange Format and Control Structures

4. Comparison of ANSI ASC X12 and UN/EDIFACT Control/Service Segments

The following subsections present a comparison of the X12 and UN/EDIFACT versions of the basic control elements of an EDI exchange. In particular, comparisons are presented for the header and trailer segments of the interchange envelope, the functional group, and the message/transaction set. Certain control structures, most notably the security headers and trailers for the functional group and the transaction set, are not compared, since they only occur in X12 and not in UN/EDIFACT. Nevertheless, where the presence of these structures leads to differing capabilities between the standards, such consequences are discussed.

4.1 Interchange Control

4.1.1 Interchange Header

The following elements are common to the interchange headers of both standards:

1. sender information -- indicates who is sending this interchange,
2. receiver information -- indicates the intended recipient of this interchange,
3. date and time -- indicates when the interchange was prepared,

4. interchange I.D. -- (referred to as the interchange control reference in EDIFACT, or the interchange control number in X12),
5. acknowledgement request -- indicates whether or not an acknowledgment is requested,
6. test indicator -- indicates whether the interchange consists of test or production data,
7. version identifier -- indicates which version of the standard contains the transaction set/message definitions used in the current interchange, (referred to as the syntax identifier in UN/EDIFACT, and the interchange control standards I.D. in X12),
8. security information -- (referred to as the recipient's reference and password in UN/EDIFACT).

The following interchange header elements are contained in the X12 standard, but not in UN/EDIFACT:

1. authorization information -- This includes additional I.D. or authorization information relating to the sender of, or the data in, the interchange. This information field may contain codes which identify the type of information in the interchange so that the data can be handled in a manner appropriate to the context. For example, the data may be defense-related information as categorized by the Department of Defense, railroad communications as categorized by the railroad industry, or telecommunications data as categorized by the communications industry.

2. component element separator -- This separator is explicitly specified in the X12 interchange, requiring two bytes for its inclusion. Because of the fixed size and mandatory nature of the X12 interchange header, the segment terminator and the segment tag/data element separator do not require separate data fields to identify them to the interchange recipient. Rather, they can be identified through normal parsing of this header. The only overhead associated with identifying the X12 separators and terminators, therefore, is the two bytes required for this component element separator data field.

   While employing the same number and types of separators (i.e., the segment terminator, the data element separator, and the subelement separator), UN/EDIFACT takes a somewhat different approach to identifying these three separators to the interchange recipient. By prior general agreement, certain character sets have been standardized, and within these character sets, specific characters have been designated to serve the functions of these separators and terminators. Therefore, all that need be conveyed to the recipient is an indication of which character set is being used, since the default separators associated with that character set will then be assumed. Five bytes are required to accomplish this information exchange, a net increase of 3 bytes over the X12 requirements.

To provide the flexibility to alter the characters used for separators, UN/EDIFACT uses the optional "UNA" data segment in which user selected characters can be designated for delimiting purposes. This flexibility comes with the overhead cost of an additional 16 bytes to accommodate the entire data segment.

The following interchange header elements are contained in the UN/EDIFACT standard, but not in X12:

The first two elements described below are related to the IA (interchange partners agreement), and the third one is unique to the UN/EDIFACT standard.

1. processing priority code -- if specified in the IA, this value is used to specify the desired priority at which the sender desires the recipient to process the interchange,

2. communications agreement I.D. -- if specified in the IA, this field is used to identify the type of communication agreement.

3. application reference -- an optional message I.D. field if the interchange contains only one type of message.

4.1.2 Interchange Trailer

There is essentially no difference between the information contained in the interchange trailer of UN/EDIFACT and that of X12. They each convey the number of included functional groups and the interchange control number.

4.1.3 General Observations Regarding The Interchange Header and Trailer

The comparison chart of record length in bytes for both mandatory and optional data elements specified for the interchange header and trailer in both standards is presented in Table 1. A study of this table indicates that the X12 approach defines a more consistently sized interchange envelope than UN/EDIFACT and that the X12 interchange envelope is larger if compared with the minimum size possible in UN/EDIFACT format. However, because of the size variability and element optionality permitted by the UN/EDIFACT specification, the UN/EDIFACT interchange envelope can equally likely be larger than the X12 interchange envelope. Because the determining factors relate to the static standard definitions, the requirements of the interchange agreements, and the specifics of the actual data used, the actual header and trailer sizes cannot be predicted a priori.

At the level of the interchange, both standards presently contain security related information. However, X12 not only provides for a password, as does UN/EDIFACT, but offers the capability of sending additional authentication information, as well. While, at present, neither standard is

generally considered to be able to support very rigorous security requirements, X12 does provide slightly increased capability at the interchange level, and significantly greater security capability at both the functional group and transaction set level. Nevertheless, functional comparability between the two standards is a reasonable expectation for the near future since additional capability is currently in the process of being designed into UN/EDIFACT.

Table 1. Comparison Of Header and Trailer Record Lengths For ANSI ASC X12 and UN/EDIFACT Interchanges

Interchanges		Mandatory Data Elements*			Optional Data Elements			Total	
		min. # of bytes	max. # of bytes	seg. tag, separators, terminator	min. # of bytes	max. # of bytes	sepa-rators	min. # of bytes	max. # of bytes
Header	X12 (ISA)	86	86	20	0	0	0	106	106
	EDIFACT (UNB)	18	99	15	12	104	7	52	225
Trailer	X12 (IEA)	10	14	6	0	0	0	16	20
	EDIFACT (UNZ)	2	20	6	0	0	0	8	26

* If a mandatory data element contains optional component data elements, these optional elements are not included in this measure. Only the mandatory component data elements are included in this measure.

4.2 Functional Group Control

4.2.1 Functional Group Header

The following functional group header elements are common to both standards:

1. functional group I.D.,
2. application sender's I.D. -- used to identify the department within the originating sender's organization,
3. application receiver's I.D. -- used to identify the department within the recipient's organization for which the group of messages is intended,
4. date and time -- indicates when the functional group was prepared,
5. functional group reference number,
6. controlling (EDIFACT) or responsible (X12) agency code -- identifies the agency that publishes and maintains the message type,

7. message version/release code-- indicates the version, release, subrelease, and industry identifier of the EDI standard being used.

The following functional group header elements are contained in the UN/EDIFACT standard, but not in X12:

EDIFACT has one additional information field in the functional group header, the "application password". This is an optional security field, of up to 14 bytes.

4.2.2 Functional Group Trailer

There is essentially no difference between the information contained in the functional group trailer of UN/EDIFACT and that of X12. They each convey the number of included transaction sets/messages and the functional group control number.

4.2.3 General Observations Regarding The Functional Group Header and Trailer

The comparison chart of record length in bytes for both mandatory and optional data elements specified for the functional group header and trailer in both standards is presented in Table 2.

It was indicated above that the UN/EDIFACT functional group header provides for a password while X12 does not. One should not infer from this, however, that X12 is lacking in this security component. In fact, it is probably more accurate to say that at the current state of the standards, X12 provides a significantly greater level of security protection at the granularity of the functional group. That is, X12 provides a security header and trailer for the functional group level which addresses issues of both authentication and encryption. Since nothing comes without a cost, these optional X12 security features understandably come at the cost of increased message size. This tradeoff, however, is generally considered worthwhile when the increased functionality is deemed necessary.

Table 2. Comparison of Header and Trailer Record Lengths For ANSI ASC X12 and UN/EDIFACT Functional Groups

Functional Groups		Mandatory Data Elements*			Optional Data Elements			Total	
		min. # of bytes	max. # of bytes	seg. tag, separators, terminator	min. # of bytes	max. # of bytes	separators	min. # of bytes	max. # of bytes
Header	X12 (GS)	19	69	11	0	0	0	30	80
	EDIFACT (UNG)	17	108	16	4	28	1	38	153
Trailer	X12 (GE)	2	15	5	0	0	0	7	20
	EDIFACT (UNE)	2	20	6	0	0	0	8	26

* If a mandatory data element contains optional component data elements, these optional elements are not included in this measure. Only the mandatory component data elements are included in this measure.

4.3 Transaction Set/Message Control

4.3.1 Transaction Set/Message Header

The following transaction set/message elements are common to both standards:

1. message reference number -- this field contains the sender's unique message reference. While this field is mandatory in both X12 and UN/EDIFACT, the size constraints are different in the two standards. X12 requires that this field be between 4 and 9 bytes long, while UN/EDIFACT specifies that the size may vary from 1 to 14 bytes in length.

2. transaction set/message I.D. -- in X12, the I.D. is a 3 byte, fixed length, mandatory field. In UN/EDIFACT, it is a segment consisting of 4 mandatory and 1 optional data elements, each of variable length. The 4 mandatory data elements convey the message type, version number, release number, and the controlling agency. The optional data element identifies the association responsible for the design and maintenance of the message type. This segment, consisting of 5 data elements, can reach a maximum total record length of 20 bytes.

The following transaction set/message data elements are contained in the UN/EDIFACT standard, but not in X12:

1. common access reference -- This optional data element serves as a key to relate subsequent transfers of data to a particular business case or file. It provides message grouping information and can vary in length from zero to 35 bytes.

2. status of the transfer -- This optional composite data element provides a message sequence (order) number within a message group. This element can be up to 3 bytes in length.

4.3.2 Transaction Set/Message Trailer

In the transaction set/message trailer, both standards contain essentially identical information: i.e., the number of included segments, and the transaction set control number (X12) or the message reference number (EDIFACT).

4.3.3 General Observations Regarding The Transaction Set/Message Header and Trailer

The comparison chart of record length in bytes for both mandatory and optional data elements specified for the transaction set/message header and trailer in both standards is presented in Table 3.

In a manner similar to that discussed in regard to functional groups, X12 provides a security header and trailer to offer both authentication and encryption at the transaction set level. At present, this level of security is not offered by UN/EDIFACT.

On the other hand, at the transaction set/message level, UN/EDIFACT provides slightly more specificity regarding the particular message types included and the status of the transfer. X12 does not provide for status information at the level of the transaction set, although status information is available at the interchange level. This status information identifies the position of the current message in a sequence of messages (i.e., first message in the sequence, final message in the sequence, or the Nth message in the sequence of messages).

Table 3. **Comparison of Header and Trailer Record Lengths For ANSI ASC X12 Transaction Sets and UN/EDIFACT Messages**

Transaction Set or Message		Mandatory Data Elements*			Optional Data Elements			Total	
		min. # of bytes	max. # of bytes	seg. tag, separators, terminator	min. # of bytes	max. # of bytes	sepa-rators	min. # of bytes	max. # of bytes
Header	X12 (ST)	7	12	5	0	0	0	12	17
	EDIFACT (UNH)	5	28	10	4	44	3	22	85
Trailer	X12 (SE)	5	19	5	0	0	0	10	24
	EDIFACT (UNT)	2	20	6	0	0	0	8	26

* If a mandatory data element contains optional component data elements, these optional elements are not included in this measure. Only the mandatory component data elements are included in this measure.

5. Conclusion

5.1 Comparison of Standards

In comparing the X12 and UN/EDIFACT EDI standards, attention was focused on the following three areas which have the potential for making a difference. First, do the two standards provide different functionality? This determination entails consideration of several issues. For example, are there functions that one standard can perform that the other cannot? Does one of the standards have the ability to represent and/or interchange certain types of information which the other standard cannot? Does the information model compelled by one standard better fit a user's needs in representing the appropriate information than the model compelled by the other standard? A second area of interest is efficiency. That is, is the nature of one of the standards such that it provides for either an advantage or disadvantage over the other standardized approach in message size, storage requirements, bandwidth utilization, or processing efficiency? And finally, it is important to look at the relative standards development processes and the existing level of standards development to assess whether one of these standards better satisfies existing user needs than the other and also whether the promise of necessary future development is realistic and can be anticipated in a timely fashion from either or both standards.

5.1.1 Functionality and Information Representation Capability Comparison

The analysis and comparison of the X12 and UN/EDIFACT standards shows that these two standards are extremely similar in structure, function, and syntax rules. There does not appear to be a type of information which can be represented in one standard which cannot be represented in the other. In fact, the structures available for transaction set/message definition are remarkably similar in both standards. Moreover, there appears to be no substantive difference in the merit of the definition rules of one standard over the other. Consequently, there is no reason to suspect that transaction sets/messages will be better or more appropriately designed by using one standard rather than the other. In both cases, the efficiency and usefulness of a defined transaction set/message relies primarily on how good a job is done by the designer. Neither standard seems to have an inherent advantage over the other in this regard. This, however, is not to say that the resultant messages from these two standardized approaches will look identical.

Whenever different individuals or groups tackle a complex problem, the likelihood of deriving identical solutions approaches zero. While X12 and UN/EDIFACT do not have any substantive differences affecting message efficiency or utility, these standards do seem to have advocates and development communities which take slightly different views of how to organize the transaction set/message information. Consequently, depending upon the orientation of the message designer, the actual transaction set/message defined by one approach, or organization, may be more closely attuned to the intended user community than that defined by the other. These differences, if they arise, can primarily be attributed to the user community, rather than to any inherent aspect of the particular EDI standard.

One difference in conceptual approach to the organization of the message information which has been incorporated in the standard syntax, however, is the way in which the concept of functional grouping is used. While a similarly named concept exists in both X12 and UN/EDIFACT, this concept is used somewhat differently and implies a slightly different organization of the message information in each of these standards. The intended purpose of this structure, as used in the X12 standard, is to provide statically defined groupings of transaction set types. These transaction sets are aggregated based upon "universal" agreement that they will commonly be used with one another. The functional group, therefore, serves to delimit groupings of these"related" transactions set types within an information interchange. In practice, it is interesting to note, most X12 functional groups consist of only a single type of transaction set. There are, presently, only a very few functional groups which can contain more than one type of transaction set.

In contrast to X12, UN/EDIFACT takes a more dynamic view of how the functional group mechanism is to be used to group related types of messages within an interchange. Rather than predefining which messages can be grouped with which other ones, UN/EDIFACT uses the functional group to specify, often on an interchange by interchange basis, which messages are packaged together. This packaging can be useful in facilitating message processing. For example, related information can be grouped together so as to conform to the division of labor in the receiving organization, thereby enabling easy dissemination of various parts of the information

interchange to the appropriate processing department in the receiving organization.

While somewhat similar, particularly in that they are both aggregating mechanisms, the difference between the functional group mechanisms in X12 and UN/EDIFACT implies a different view, and organization, of the information by the two different approaches. Therefore, although both standards enable essentially the same things to be done in organizing the information, there will most likely be differences in the defined transaction sets/messages depending upon the orientation of the definer.

5.1.2 Efficiency Comparison

Some slight differences, attributable to the overhead of control structures in the interchanges, do exist between these two approaches. For example, the data elements comprising the control segments in X12 are small in number, mostly fixed in length, and mandatory for inclusion. UN/EDIFACT, by contrast, provides a greater number of control segment elements, allows variability in the lengths of these elements, and makes most of these additional control segment elements optional for use.

These differences in characteristics affect two aspects of the information interchange -- i.e., the size of the control segments of the interchange message, and the ease of parsing them. Regarding the size of the interchange, while the maximum data length of a UN/EDIFACT transaction attributable to the control segments can be longer than the fixed data length specified in X12, in actual usage, the minimum data length specified in UN/EDIFACT's control segments can and, perhaps, often will be shorter than that in X12. The variable length and optional use of these elements allows UN/EDIFACT interchanges to use only the number of bytes necessary to convey the intended information and to customize the message size to the message contents. This can offer economies of reduced data transmission bandwidth and data storage requirements. There is, however, a trade-off for this flexibility. The cost incurred is that variable length records require more complex and time-consuming processing than the fixed length records of X12. In view of these considerations, there is no clear cut size advantage that one approach has over the other. The actual size differential of the interchange, because it is dependent upon the contributions made by both the control segments and the message segments, can favor X12 for one message type and UN/EDIFACT for a different message type.

5.1.3 Existing Level of Standard Development and Anticipated Progress Comparison

Perhaps because of a headstart in the development process, or possibly because of a more homogeneous development group enabling more rapid consensus, there are features which currently exist in the X12 standard which have not as yet been provided in the UN/EDIFACT standard. In particular, there are security features and optional acknowledgement capabilities for interchanges and functional groups. Moreover, X12, unlike UN/EDIFACT, addresses the issue of the interchange of binary data and provides a structure by which to accomplish such an interchange. The lack of these features in the UN/EDIFACT standard is most likely attributable

to the relative youth of that standard compared with the X12 standard. What will evolve in the UN/EDIFACT standard to address these needs, however, is still to be determined.

At the present time, for areas such as health care, X12 has a clear lead in defining transaction sets. Since there are almost no currently defined UN/EDIFACT messages for health care needs, X12 provides the only currently available standardized solution to EDI for health care applications. As with the other issues mentioned above, the availability of health care related messages in UN/EDIFACT is a question to be resolved in the future and will be greatly affected by whatever X12-to-UN/EDIFACT migration strategy is eventually adopted.

5.2 ANSI ASC X12 to UN/EDIFACT Migration

Recognizing that the X12 and UN/EDIFACT are concurrently developing divergent EDI standards, X12 members voted in 1992 to "adopt a single EDI standard which is EDIFACT, after the release of version 4 of the X12 ANSI standard which is expected in 1997." However, the current draft X12 migration plan [7] developed by the X12 Steering Committee (which, at the time of the writing of this paper, has just been adopted), modifies that decision to allow the parallel development of X12 and UN/EDIFACT for an indefinite period to be determined by the X12 community. The importance of this issue relates to the basic reason for attempting to develop these EDI standards in the first place. Providing a single, standardized protocol and set of message definitions enables users to communicate in "the same language" and interoperate. Having more than one standard set of messages subverts that single language approach and complicates communication. Alignment of the data element, segment, and transaction set/message dictionaries will greatly simplify EDI communication. In view of this, a migration plan to move from having two standardized approaches to having only one standardized solution is the most desirable course.

Short of achieving agreement on a single method and set of messages, a strategy for converting from one format to the other format will be needed. However, when conversion or translation is employed, there is generally some loss of semantic information, because the source and target formats are not fully comparable. In addition, there is increased overhead encountered in doing the conversion.

Whether the U.S. government should endorse UN/EDIFACT for its domestic EDI interchanges with particular businesses within particular industries may depend upon what makes business sense for both the industry and government partners participating in those interchanges[8]. The latest EDI FIPS [9] shows no preference in domestic interchanges (e.g., for X12), but does show a preference for international interchanges to use UN/EDIFACT.

Unfortunately, there is often a disconnect between the focus and decisions of the standards committees and what users, such as business, industry, and government need and want to use. Consequently, if a particular standard doesn't meet the needs of a particular user group, mandating its use serves no useful purpose. Therefore, the government may reasonably attempt

to assess commonality of need and utility, and may legitimately urge conformity to a single approach when reasonable and, more importantly, when agreed to by users. But, it is generally ineffective and counter-productive to mandate uniformity merely for the sake of uniformity when users will not be in agreement about the utility of a given standardized approach.

In contemplating possible ways to ameliorate the multiple standards situation, particularly in the case of X12 and UN/EDIFACT which we have just seen to be extremely similar in many regards, one rational approach does come to mind. In view of the considerable similarity between the X12 and UN/EDIFACT approaches, and since the division of labor between the two groups does tend to dilute the effort, it would seem quite reasonable that a grand unification of the standards should legitimately take place, thus leading to greater progress and interoperability. Furthermore, and perhaps most controversially, it seems that such a unified effort can most reasonably be justified to be sponsored by UN/EDIFACT since it enjoys the more global mandate. However, this does not mean that there should be a wholesale discarding of previous work. Rather, there are considerable gains to be realized by making use of already developed solutions to problems which the longer standing X12 work effort has achieved. For example, in the case of health care transaction sets, while there are none defined as yet in UN/EDIFACT, there is a rather comprehensive set of them already defined in X12. If the "not designed here" prejudice could be overcome, and the X12 transaction sets could be offered up and accepted by UN/EDIFACT to be converted/translated into UN/EDIFACT form and then adopted wholesale, this would provide a tremendous boost in the coverage of the UN/EDIFACT standard as well as provide considerable movement toward embracing a single standard. Certainly, a reasonable and useful action to be taken by those parties interested in reducing redundancy and/or non-interoperability caused by multiple standards would be to lobby the appropriate standards bodies to effect this convergence of these major EDI standards toward a unified approach. Toward this end, in their alignment plan [7], X12 has called for the development of equivalent business functionality in UN/EDIFACT to facilitate migration to the singular usage of UN/EDIFACT.

6. REFERENCES

[1] Electronic Data Interchange Standards, Draft Version 3 Release 4 Subrelease 2 --Volume I: Control Standards, Transaction Set Tables, and Segment Directory -- American National Standards Institute (ANSI) Accredited Standards Committee (ASC) X12 , June 1994.

[2] Electronic Data Interchange Standards, Draft Version 3 Release 4 Subrelease 2 --Volume II: Data Element Dictionary -- American National Standards Institute (ANSI) Accredited Standards Committee (ASC) X12, June 1994.

[3] Draft Messages and Directories, Version D93.A -- Volume I: Supplemental Documentation and Directories -- United Nations Electronic Data Interchange For Administration, Commerce, and Transport (UN/EDIFACT), May 1994.

[4] Draft Messages and Directories, Version D93.A -- Volume II: Standard Messages BANSTA through IFTMBC -- United Nations Electronic Data Interchange For Administration, Commerce, and Transport (UN/EDIFACT), May 1994.

[5] Draft Messages and Directories, Version D93.A -- Volume III: Standard Messages IFTMBF through SUPMAN -- United Nations Electronic Data Interchange For Administration, Commerce, and Transport (UN/EDIFACT), May 1994.

[6] Stephen Luster, Lisa Janssen, and Kathleen Miller, "Department of Defense Use of UN/EDIFACT Standards...A Readiness Study and Migration Strategy", PL311R1, prepared pursuant to Department of Defense Contract MDA903-90-C-0006 by Logistics Management Institute, May 1994.

[7] ASC X12 Plan for Technical Migration To and Administrative Alignment With UN/EDIFACT, American National Standards Institute (ANSI) Accredited Standards Committee (ASC) X12, approved January 13, 1995, modified February 6, 1995.

[8] R. G. Saltman, "Standardization of Electronic Data Interchange - U.S. Government Activities", The DataLaw Report, July 1994, pp. 3-26.

[9] FIPS PUB 161-1, Federal Information Processing Standards Publication -- Electronic Data Interchange (EDI), National Institute of Standards and Technology (NIST), 19 April 1993.